CLAIMS

What is claimed is:

1. In an electronic device, a method for management of software, comprising the steps of:

determining a functionality of a unit of code and an environment configuration suitable for executing said unit of code; and

automatically providing a file name corresponding to said functionality for said unit of code.

2. The device of claim 1, the method further comprising the step of locating a file having said file name in a directory corresponding to said environment configuration.

3. The device of claim 2, the method further comprising the step of naming said directory to have a directory name corresponding to said environment configuration.

4. The device of claim 2, wherein a plurality of characteristics of said environment configuration include at least one of the group of a word size on a target processor, a word size on a host processor, an execution software type, an execution software version number and an operating system.

5. The device of claim 3, wherein said naming step determines said directory name by the use of a checksum of a plurality of characteristics of said environment configuration.

6. The device of claim 3, wherein said characteristics of said environment configuration include at least one of the group of a word size on a target processor, a word size on a host processor, an execution software type, an execution software version number and an operating system.

7. The device of claim 1, wherein said file name is determined by the use of a checksum of characteristics of said functionality.

8. The device of claim 1, wherein said checksum is based on at least one of the group of a MD5 checksum and a CRC checksum.

9. The device of claim 7, wherein said characteristics of said functionality include at least one of the group of an input type and an output type of said unit of code, an operation on an input to said unit of code.

10. The device of claim 1, wherein said step of automatically providing a file name provides said file name also corresponding to said environment configuration.

11. The device of claim 1, wherein said step of automatically providing a file name determines said file name by the use of a checksum of a plurality of characteristics of said functionality and said environment configuration.

12. The device of claim 1, wherein said step of automatically providing a file name determines said file name by the use of a consistent naming scheme representative of characteristics of said functionality.

13. The device of claim 1, wherein said file name is comprised of characters pertaining to an input type and an output type of said unit of code.

14. The device of claim 1, wherein said file name is comprised of characters pertaining to said functionality of said unit of code.

15. The device of claim 1, wherein said file name also corresponds to said environment configuration.

16. The device of claim 1, wherein said unit of code is representative of a portion of a block diagram environment.

17. The device of claim 1, wherein said unit of code is representative of a portion of a modeling environment.

18. In an electronic device, a method for management of software, comprising the steps of:

determining a functionality of a unit of code and an environment configuration suitable for executing said unit of code; and

automatically providing a function name corresponding to said functionality for said unit of code.

19. The device of claim 18, the method further comprising the step of locating a function having said function name in a constituent of an organizational structure corresponding to said environment configuration.

20. The device of claim 19, the method further comprising the step of naming said constituent to have a constituent name corresponding to said environment configuration.

21. The device of claim 20, wherein a plurality of characteristics of said environment configuration include at least one of the group of a word size on a target processor, a word size on a host processor, an execution software type, an execution software version number and an operating system.

22. The device of claim 20, wherein said naming step determines said constituent name by the use of a checksum of a plurality of characteristics of said environment configuration.

23. The device of claim 22, wherein said characteristics of said environment configuration include at least one of the group of a word size on a target processor, a word size on a host processor, an execution software type, an execution software version number and an operating system.

24. The device of claim 19, wherein said function name is determined by the use of a checksum of characteristics of said functionality.

25. The device of claim 19, wherein said checksum is based on at least one of the group of a MD5 checksum and a CRC checksum.

26. The device of claim 24, wherein said characteristics of said functionality include at least one of the group of an input type and an output type of said unit of code, an operation on an input to said unit of code.

27. The device of claim 19, wherein said step of automatically providing a function name provides said function name also corresponding to said environment configuration.

28. The device of claim 19, wherein said step of automatically providing a function name determines said function name by the use of a checksum of a plurality of characteristics of said functionality and said environment configuration.

29. The device of claim 19, wherein said step of automatically providing a function name determines said function name by the use of a consistent naming scheme representative of characteristics of said functionality.

30. The device of claim 19, wherein said function name is comprised of characters pertaining to an input type and an output type of said unit of code.

31. The device of claim 19, wherein said function name is comprised of characters pertaining to said functionality of said unit of code.

32. The device of claim 19, wherein said function name also corresponds to said environment configuration.

33. The device of claim 19, wherein said unit of code is representative of a portion of a block diagram environment.

34. The device of claim 19, wherein said unit of code is representative of a portion of a modeling environment.

35. In an electronic device, a method for management of software, comprising the steps of:

determining a functionality of a unit of code and an environment configuration suitable for executing said unit of code; and

automatically providing a macro name corresponding to said functionality for said unit of code.

36. The device of claim 35, the method further comprising the step of locating a macro having said macro name in a constituent of an organizational structure corresponding to said environment configuration.

37. The device of claim 36, the method further comprising the step of naming said constituent to have a constituent name corresponding to said environment configuration.

38. The device of claim 37, wherein a plurality of characteristics of said environment configuration include at least one of the group of a word size on a target processor, a

word size on a host processor, an execution software type, an execution software version number and an operating system.

39. The device of claim 37, wherein said naming step determines said constituent name by the use of a checksum of a plurality of characteristics of said environment configuration.

40. The device of claim 39, wherein said characteristics of said environment configuration include at least one of the group of a word size on a target processor, a word size on a host processor, an execution software type, an execution software version number and an operating system.

41. The device of claim 37, wherein said macro name is determined by the use of a checksum of characteristics of said functionality.

42. The device of claim 37, wherein said checksum is based on at least one of the group of a MD5 checksum and a CRC checksum.

43. The device of claim 41, wherein said characteristics of said functionality include at least one of the group of an input type and an output type of said unit of code, an operation on an input to said unit of code.

44. The device of claim 37, wherein said step of automatically providing a macro name provides said macro name also corresponding to said environment configuration.

45. The device of claim 37, wherein said step of automatically providing a macro name determines said macro name by the use of a checksum of a plurality of characteristics of said functionality and said environment configuration.

46. The device of claim 37, wherein said step of automatically providing a macro name determines said macro name by the use of a consistent naming scheme representative of characteristics of said functionality.

47. The device of claim 37, wherein said macro name is comprised of characters pertaining to an input type and an output type of said unit of code.

48. The device of claim 37, wherein said macro name is comprised of characters pertaining to said functionality of said unit of code.

49. The device of claim 37, wherein said macro name also corresponds to said environment configuration.

50. The device of claim 37, wherein said unit of code is representative of a portion of a block diagram environment.

51. The device of claim 37, wherein said unit of code is representative of a portion of a modeling environment.

52. In an electronic device, a method for management of software, comprising the steps of:

determining a functionality of a unit of code and an environment configuration suitable for executing said unit of code; and

automatically providing a class name corresponding to said functionality for said unit of code.

53. The device of claim 52, the method further comprising the step of locating a class having said class name in a constituent of an organizational structure corresponding to said environment configuration.

54. The device of claim 53, the method further comprising the step of naming said constituent to have a constituent name corresponding to said environment configuration.

55. The device of claim 54, wherein a plurality of characteristics of said environment configuration include at least one of the group of a word size on a target processor, a word size on a host processor, an execution software type, an execution software version number and an operating system.

56. The device of claim 54, wherein said naming step determines said constituent name by the use of a checksum of a plurality of characteristics of said environment configuration.

57. The device of claim 56, wherein said characteristics of said environment configuration include at least one of the group of a word size on a target processor, a word size on a host processor, an execution software type, an execution software version number and an operating system.

58. The device of claim 52, wherein said class name is determined by the use of a checksum of characteristics of said functionality.

59. The device of claim 52, wherein said checksum is based on at least one of the group of a MD5 checksum and a CRC checksum.

60. The device of claim 58, wherein said characteristics of said functionality include at least one of the group of an input type and an output type of said unit of code, an operation on an input to said unit of code.

61. The device of claim 52, wherein said step of automatically providing a class name provides said class name also corresponding to said environment configuration.

62. The device of claim 52, wherein said step of automatically providing a class name determines said class name by the use of a checksum of a plurality of characteristics of said functionality and said environment configuration.

63. The device of claim 52, wherein said step of automatically providing a class name determines said class name by the use of a consistent naming scheme representative of characteristics of said functionality.

64. The device of claim 52, wherein said class name is comprised of characters pertaining to an input type and an output type of said unit of code.

65. The device of claim 52, wherein said class name is comprised of characters pertaining to said functionality of said unit of code.

66. The device of claim 52, wherein said class name also corresponds to said environment configuration.

67. The device of claim 52, wherein said unit of code is representative of a portion of a block diagram environment.

68. The device of claim 52, wherein said unit of code is representative of a portion of a modeling environment.

69. In an electronic device, a method for management of software, comprising the steps of:

providing an organizational structure having a plurality of constituents, said constituents corresponding to unique environment configurations; and

providing an identifier corresponding to a functionality of a unit of code in said organizational structure.

70. The device of claim 69, the method further comprising the step of locating said unit of code having said identifier in a constituent, of said plurality of constituents corresponding to said environment configuration.

71. The device of claim 70, the method further comprising the step of naming said constituent to have a constituent name corresponding to said environment configuration.

72. The device of claim 71, wherein said naming step determines said constituent name by the use of a checksum of a plurality of characteristics of said environment configuration.

73. The device of claim 69, wherein said organizational structure is a directory structure and said identifier is a file name.

74. In an electronic device, a method for management of software, comprising the steps of:

selecting a utility to process having a first functionality;

determining a characteristic of a first environment configuration suitable for operation of said utility;

searching an organizational structure for a constituent corresponding to said first environment configuration;

creating said constituent corresponding to said first environment configuration, if said constituent corresponding to said first environment configuration is not found in said searching an organizational structure step;

generate a name for said first functionality;

searching said constituent corresponding to said first environment configuration for an identifier of a unit of code, said identifier corresponding to said name for said first functionality; and

create said unit of code having said first functionality and suitable for execution in said first environment configuration, if said identifier is not found in said searching said constituent corresponding to said first environment configuration step.

75. The device of claim 74, wherein said searching an organizational structure step uses a checksum of a plurality of characteristics of said first environment configuration.

76. The device of claim 74, wherein said step of searching a constituent corresponding to said first environment configuration uses a checksum of a plurality of characteristics of said first functionality.

77. The device of claim 76, the method further comprising the steps of:

generating a comment string corresponding to said first functionality of said unit of code; and

verifying said functionality of said unit of code by the use of said checksum and said comment string.

78. The device of claim 74, wherein said utility is operational in a block diagram environment.

79. The device of claim 74, wherein said utility is operational in a modeling environment.

80. The device of claim 74, wherein said identifier is a file name.

81. The device of claim 74, wherein said identifier is a function name.

82. The device of claim 74, wherein said identifier is a macro name.

83. The device of claim 74, wherein said identifier is a class name.

84. The device of claim 74, wherein said organizational structure is a directory structure.

85. The device of claim 74, wherein said organizational structure is a class structure.

86. The device of claim 74, wherein said constituent is a file.

87. A computer readable medium containing a software tool for executing a method for management of software, comprising the steps of:

selecting a utility to process having a first functionality;

determining a characteristic of a first environment configuration suitable for operation of said utility;

searching an organizational structure for a constituent corresponding to said first environment configuration;

creating said constituent corresponding to said first environment configuration, if said constituent corresponding to said first environment configuration is not found in said searching an organizational structure step;

generate a name for said first functionality;

searching said constituent corresponding to said first environment configuration for an identifier of a unit of code, said identifier corresponding to said name for said first functionality; and

create said unit of code having said first functionality and suitable for execution in said first environment configuration, if said file name is not found in said searching said constituent corresponding to said first environment configuration step.

88. The device of claim 87, wherein said searching an organizational structure step uses a checksum of a plurality of characteristics of said first environment configuration.

89. The device of claim 87, wherein said step of searching a constituent corresponding to said first environment configuration uses a checksum of a plurality of characteristics of said first functionality.

90. The device of claim 89, the method further comprising the steps of:

generating a comment string corresponding to said first functionality of said unit of code; and

verifying said functionality of said unit of code by the use of said checksum and said comment string.

91. The device of claim 87, wherein said utility is operational in a block diagram environment.

92. The device of claim 87, wherein said utility is operational in a modeling environment.

93. The device of claim 87, wherein said identifier is a file name.

94. The device of claim 87, wherein said identifier is a function name.

95. The device of claim 87, wherein said identifier is a macro name.

96. The device of claim 87, wherein said identifier is a class name.

97. The device of claim 87, wherein said organizational structure is a directory structure.

98. The device of claim 87, wherein said organizational structure is a class structure.

99. The device of claim 87, wherein said constituent is a file.